Understanding Deep Neural Networks For Regression In Leaf Counting

Andrei Dobrescu
University of Edinburgh
A.Dobrescu@ed.ac.uk

Mario Valerio Giuffrida
University of Edinburgh
V.Giuffrida@ed.ac.uk

Sotirios A. Tsaftaris
University of Edinburgh
The Alan Turing Institute
S.Tsaftaris@ed.ac.uk

Abstract

Deep learning methods are constantly increasing in popularity and success across a wide range of computer vision applications. However, they are perceived as 'black boxes', due to the lack of an intuitive interpretation of their decision processes. We present a study aimed at understanding how Deep Neural Networks (DNN) reach a decision in regression tasks. This study focuses on deep learning approaches in the common plant phenotyping task of leaf counting. We employ Layerwise Relevance Propagation (LRP) and Guided Back Propagation to provide insight into which parts of the input contribute to intermediate layers and the output. We observe that the network largely disregards the background and focuses on the plant during training. More importantly, we found that the leaf blade edges are the most relevant part of the plant for the network model in the counting task. Results are evaluated using a VGG-16 deep neural network on the CVPPP 2017 Leaf Counting Challenge dataset.

1. Introduction

As deep learning becomes more widespread in computer vision, there is a growing need to understand the underlying decision making process of deep neural networks (DNN). However, deep architectures are typically seen as 'black boxes', lacking a straightforward explanation of how a network achieves a prediction.

The lack of insight is due to the non-linearity of the mappings that take as input raw image pixels and transform them into a feature representation from which a final classifier or regression function is computed [4]. This can be a major drawback, as it makes difficult for scientists to thoroughly verify the predicted decision. Oftentimes, the predicted output is either a pre-defined class choice for classification or a number when performing regression analysis. However, we cannot extrapolate what are the important parts of an image that contribute the most to the final result.

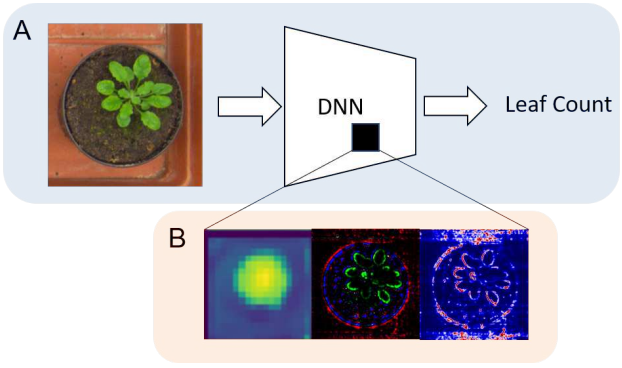


Figure 1. A: Common deep learning framework taking an image as input into a trained deep neural network (DNN) which outputs the leaf count. B: We investigate what elements of the input image contribute the most in computing a prediction and gain an understanding of the intermediate layers.

The visualization of the inner mechanism of a deep network provides a human level understanding of how the deep learning models make decisions and what image representations they have learned. Investigating the 'black box' will increase the confidence in deep learning predictions [19] and will permit the redesign of the architecture to improve performance.

Improvements in plant phenotyping are essential for the development of the sector, as they are necessary for increasing plant productivity and resistance. Leaf count is a fundamental plant trait that underlines plant development and growth stages [7]. There have been numerous studies that propose deep learning approaches in leaf counting [2, 10, 12, 15], but they have failed to answer a key question: *what does a deep neural network focus on to predict a specific leaf count?* Furthermore, the best reported results for this task have been through using a direct regression approach [10, 15], which is poorly studied from a visualization perspective. An attempt to gauge salient regions in images has been done in [1, 10], however not in much detail. Gaining insights into what contributes to the decision making process in direct regression deep learning approaches

would lead to a better understanding of the model. Additionally, visualizing what the networks focus on may help in improving the results. This can be done by choosing different methods of acquiring images, pre-processing data, or choosing better suited architectures or hyper parameters to the model.

In this paper, we aim to visualize and understand DNNs for regression problems, by finding which image areas contribute the most to leaf count. There have been several studies that describe approaches for classification [9, 14, 33], but there is a lack of experimental examples that focus on regression models. We focus on two popular methods for visualization: Layerwise Relevance Propagation (LRP) [4] and Guided Back Propagation [29]. We chose them because they are widely used and have the advantage of being able to traverse convolutional layers, which are used to extract features, as well as fully connected layers, which are then used to implement the leaf count regressor.

Our contributions are the following:

- We focus on visualizing methods for regression problems. Using the layerwise relevance, we study the influence of each input both on the final predicted value as well as in intermediate layers.
- We show that regression value is predicted mainly using the foreground object (i.e. the plant).
- We demonstrate experimentally that edges of leaf blades are the most influential for count. We found that the petiole or the central parts of leaves are not taken into account in the decision making.

In this study we aim for a functional understanding of the model, as opposed to a lower level algorithmic understanding of it. We focus on interpreting the outputs of a DNN and explain individual predictions and where the predictions come from. The main network used in this study is the VGG-16 model [28] modified to have a leaf count regressor at the top (c.f. Figure 1).

2. Related Works

When relating to DNNs, the concept of model understanding has been defined in terms of *interpretability* and *explainability* [19, 22]: an interpretation is the mapping of an abstract concept (e.g a predicted class) into a high-level domain that a human can make sense of, and an explanation is the collection of features of the interpretable domain that have contributed to a decision such as classification or regression [22]. For example, an explanation can be a heatmap highlighting specific pixels from the input, or a feature map that contributes the most to an output decision [17, 27]. In recent years, researchers have investigated a variety of approaches to visualize the inner workings of deep neural networks in computer vision problems [9, 14, 33].

2.1. Explainability

Zeiler and Fergus [31] proposed a deconvolutional network (Deconvnet) that records the activations of a convolutional layer and reverses the forward pass to display which visual patterns from the input image contribute to the observed activations. Another approach [27] uses information from lower layers in concordance with the input image to estimate the image regions responsible for activations seen at top layers. Gradient based approaches that compute gradients of a part of a DNN with relation to the input image [20] are also generally accepted visualization methods. Recently, this work was expanded in [23] by including mid level elements to identify relevant features encoded and use strided operations when performing the backwards pass to reduce visual artifacts.

In [34], the authors propose Class Activation Map using global average pooling of activations of the filters in the last convolutional layer. This yields a weighted sum over the spatial locations of the activations resulting in a class activation map which is up-sampled to the size of the input image. Grad-cam [26] and the extension Grad-cam++ [8] are visualization methods that compute weights of activation maps of trained models at layer and neuron level, achieving improved object localization in the resulting heatmaps.

2.2. Interpretation

A common method of interpretation is to cover part of the input image and to measure the difference in activations of a trained model. This method of visualization works on the assumption that occluding relevant parts of the input will lead to a significant drop in performance [31, 34]. However, the resolution of this method depends on the size of the occluded regions. Escorcia et al. [11] proposed a method to predict object attributes using a feature selection process that combines neuron activations with object categories. Similarly, other methods that match the activations of convolutional filters against a particular dataset with pixel-wise annotations have been proposed [5, 32]. PatternNet [16] has been proposed as a method to explain the contributions of the input data signal. The technique trains a linear signal estimator on top of a pre-trained DNN to explain the relation between the data signal and the attributed patterns learned by the DNN.

All the studies mentioned in this section experimentally tested their proposed methods on classification tasks, while in our study we focus instead on using visualization techniques for a regression application.

Recently, there have been studies that include some interpretation techniques in plant phenotyping. In [10], the network was still considered as a 'black box', but the authors found out that the network was mostly focusing on the plant by masking parts of the image with a black square and evaluating the impact on predictions. In [1], the authors

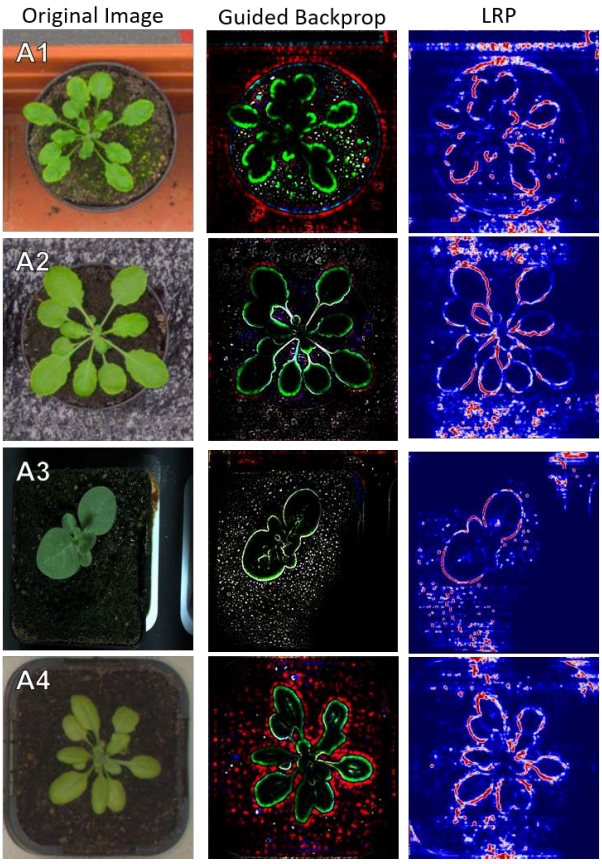


Figure 2. Visualization techniques that show which parts of the input are taken into account for a final leaf count prediction. The rows represent the four datasets available in the CVPPP 2017 Leaf Counting Challenge denoted A1-A4 containing Arabidopsis and Tobacco plants. Guided back-propagation computes positive gradients from the leaf count prediction to the input image and outputs an RGB image. Layerwise relevance (LRP) is computed using the $\alpha 2 \beta 1$ propagation rule described in section 3.2. Red indicates areas of positive relevance while blue areas represent negative relevance. The results of both techniques reveal that the most important parts of the plant are the leaf blade edges, independent of the different datasets and plant species.

use the method described in [34] to find salient regions for wheat shoot count. The drawback of this approach is that it does not work with fully connected layers. In [18], they investigate the visualization of learned leaf features for plant classification.

3. Visualization techniques

In this section we describe the visualization techniques used in this study. In particular, we provide insights on guided backpropagation [29] and layerwise relevance propagation [4]. Overall, we chose to use both of these visualization methods, as they can effectively work for convolutional and fully-connected layers. In addition, they allow intermediate layer investigation.

3.1. Guided backpropagation

This approach is a gradient-based visualization technique designed to highlight what parts of the input contribute to a given neuron in a neural network [29]. The method back propagates the gradient with relation to the input image while masking negative values. This results in only positive gradients being conserved. The advantage of retaining only positive gradients is to prevent a backward flow of negative signals corresponding to neurons which inhibit activation of the higher level neuron. As opposed to usual backpropagation, this can act as an additional guidance signal when traversing the network. The output of guided backpropagation is an RGB image of the same dimensions as the input image. This method works for visualizing neurons in convolutional as well as fully connected layers. Examples of outputs can be seen in Figure 2.

3.2. Layerwise relevance propagation (LRP)

LRP [4] is a backwards propagation technique, designed as a method of explainability in deep neural networks. It was found to be widely applicable to classification problems [3, 22]. The principle behind this technique is the conservation property: each neuron receives a share of the network output and redistributes it to its predecessors in equal amount, until the input is reached. The output of the LRP technique is a relevance heatmap highlighting which areas of an input contributes to the output. There are several advantages to this technique: firstly, it works with convolutional layers as well as fully connected layers. Secondly, it produces a heatmap that relates to the input image. Furthermore, as opposed to the guided backpropagation approach, it can capture both positive and negative evidence. Examples of generated relevance heatmaps are displayed in Figure 2.

LRP is divided in two phases. The first phase is a standard forward pass through the network, which records the activations at each layer. In the second phase, the score obtained at the output of the network is back propagated in the network adhering to propagation rules based on the relevance conservation property [4]. The rules are described as follows: let j and k be indices of neurons in two successive layers and P_k be the relevance of neuron k for the prediction of $f(x)$. Then, the term $P_{j \leftarrow k}$ is defined as the share of P_k that is redistributed to neuron j in the lower layer. The conservation property of the neuron dictates $\sum_j P_{j \leftarrow k} = P_k$ when moving towards a higher layer. Similarly, the notation of neurons from the lower level can be defined as an aggregate of the relevance corresponding to neurons in a higher layer $P_j = \sum_k P_{j \leftarrow k}$. Combining the two notations shows that the conservation of relevance holds between layers and

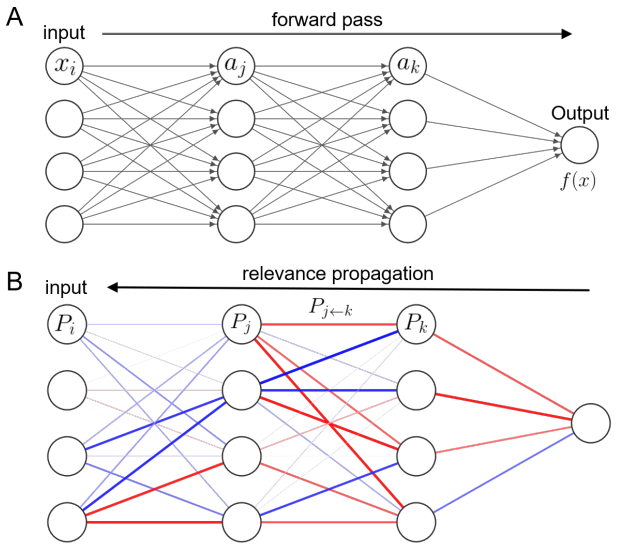


Figure 3. Diagram showing the LRP phases. A) Standard forward pass through the neural network from the input to the output. In this phase, the activations are recorded, and are shown here as a_j and a_k. B) The relevance propagation is done going backwards through the network from the output to the input. As the signal travels, it detects the relevance of nodes $P_{i\leftarrow k}$. The red and blue lines denote an excitatory or inhibitory influence, respectively.

travels from the input to the output:

$$\sum_{i=1}^{d} P_i = \ldots = \sum_j P_j = \sum_k P_k = \ldots = f(x) \qquad (1)$$

When looking at how relevance is carried over individual neurons, the conservation principle still holds and relevance can be divided into positive (excitatory influence) and negative (inhibitory influence). Let a neuron in a deep neural network be described by the following equation:

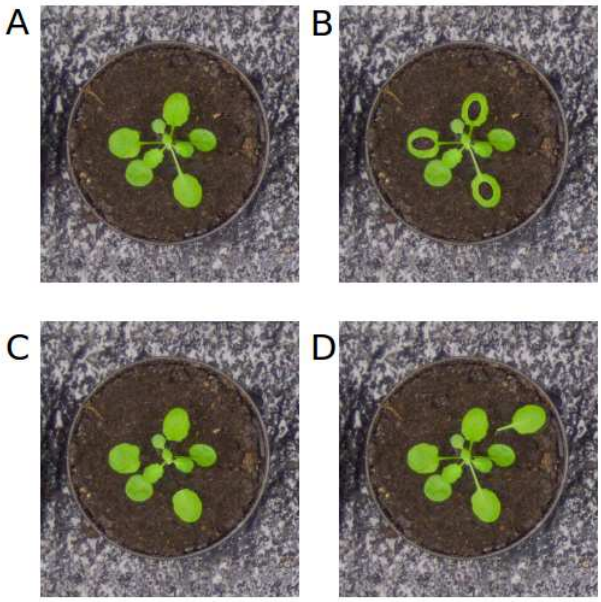
$$a_k = \varphi\left(\sum_j a_j w_{jk} + b_k\right), \qquad (2)$$

where a_k represents the activation of the neuron, a_j the activation of the previous layer, w_{jk} the weights and b_k the bias of the neuron. We assume that the function φ is a positive and monotonically increasing activation function.

The relevance signal is broken down into two factors for positive and negative relevance influence and it can be formally written as:

$$P_j = \sum_k \left(\alpha \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} - \beta \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-}\right) P_k, \qquad (3)$$

where the α and β coefficients indicate the strength of the positive and negative relevance signals to be displayed in the output heatmap. For example when $\alpha = 1$ and $\beta = 0$ only positive relevance is displayed and can be an interpretation of the deep Taylor decomposition described in [24]. An experimentally inferred good choice for the coefficient values is $\alpha = 2$ and $\beta = 1$ [4] which are the values we use for our experiments. A diagram depicting the phases of LRP and how each node contributes to the final outcome can be seen in Figure 3.



Label	A	B	C	D
Predictions	9	9	9	10

Figure 4. Test to evaluate our findings that the leaf blade edge is the most important for computing count. A: The original image with a ground truth of 9 leaves, B: The centres of several leaves have been deleted and replaced with soil texture, C: the petioles from several larger leafs have been removed altogether, D: An extra leaf has been added but not attached to the centre of the plant. The network predicts the same leaf count for images such as B and C as does for the original image meaning that no crucial information was lost. In D the network successfully detects an additional leaf even if not attached to the central plant.

4. Results

In this section, we show our findings in interpreting a deep neural network in the context of the regression task. We adopted VGG-16 [28], where the last two layers were adapted to accommodate the regression task, and maintained the same training parameters as in [10, 12]. We opted to use a VGG-16 as opposed to a ResNet-50 [13] or Inception-V3 [30] because they are not sequential networks and are generally difficult to interpret because of the presence of skip connections and residual blocks [9]. The model was trained on the CVPPP 2017 Leaf Counting Chal-

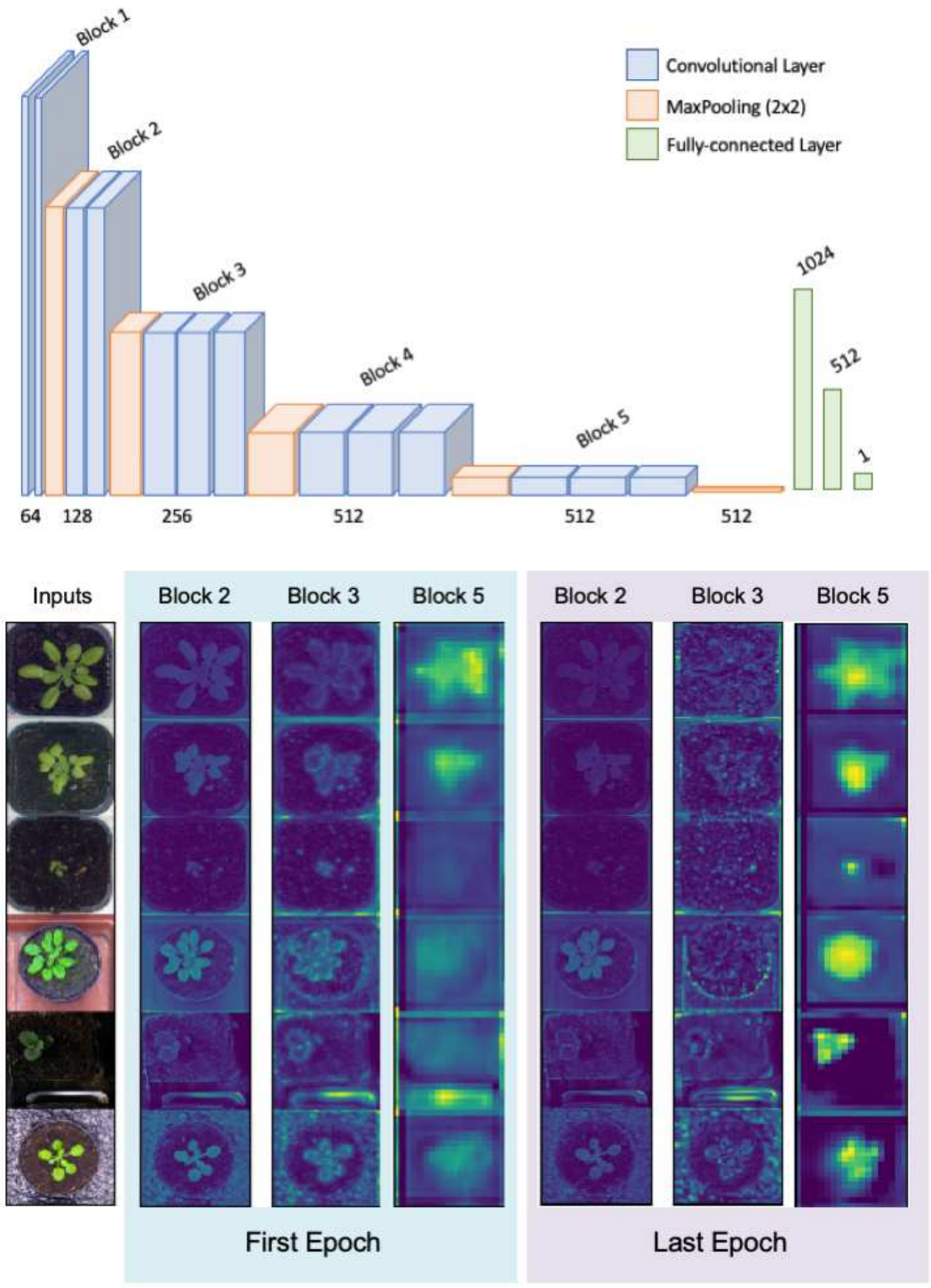


Figure 5. The top part of the figure consists of a diagram of the VGG-16 network architecture used in this study. The leaf count regressor composed of three fully connected layers was added after the convolutional blocks. The bottom part of the figure shows the average activations at the end of several convolutional blocks at the start and end of training. In the end of block 5 the areas corresponding to the plants show increased activity at the end of training for all the scales and species of plants.

Network	DiC	|DiC|	MSE	%
ResNet-50 [10]	0.23 (1.02)	0.69 (0.73)	1.11	44
VGG-16 (Ours)	0.11 (1.10)	0.74 (0.80)	1.20	41

Table 1. Comparison of VGG-16 vs. ResNet-50 leaf counting performance trained on CVPPP 2017 dataset.

lenge datasets [6, 21, 25] to perform the leaf counting task. The VGG-16 based network has similar performance to the ResNet-50 in [10], as reported in Table 1.

4.1. What is the network looking at?

In [10, 12], it was shown that the network can learn to exclude the background and focuses mostly on the plant area. However, the mentioned studies do not demonstrate which part of the plant contributes the most to the leaf counting.

In Figure 2, we show the qualitative results of the guided backpropagation (Figure 2(B)) and LRP (Figure 2(C)). The guided backpropagation method helps in understanding which parts of the input contribute to the final prediction. On the other hand, LRP is used to determine which of the highlighted areas have positive influence (coloured in red), and which have negative influence (coloured in blue) on the final prediction.

It is evident from these results that the most active parts contributing the leaf count are the leaf blade edges. We can hypothesize that neither the blade center nor the petiole have any impact in reaching the final leaf count prediction. To demonstrate this claim experimentally, we selected an image from the test dataset and we manually removed the centers of some leaves. In parallel, we similarly manipulated the image to remove the petiole of some leaves. In addition, we created another image where we manually added an extra leaf. We used our trained network to predict the leaf count on the manipulated images. In Figure 4, we show an example of the results of this experiment. We selected an image from the A2 testing dataset (Figure 4(A)), which we know has the ground truth leaf count of 9. Then we removed the inner central part of some leaves (Figure 4(B)) and the network prediction remained 9. Then we removed the petioles of several leaves (Figure 4(C)) and the total leaf count prediction was still 9. Lastly, the network was able to count 10 when an extra leaf was added to the image (Figure 4(D)). We can see from this experiment that the most relevant plant areas are the blade edges.

4.2. Intermediate Layers Analysis

Next, we analyzed the intermediate layers of the network to understand how each convolutional block in the VGG-16 processes the input data. In the top part of Figure 5, we show a representation of the VGG-16 network: it consists of 5 convolutional blocks, separated by max-pooling operations. We considered the last convolutional layer of each

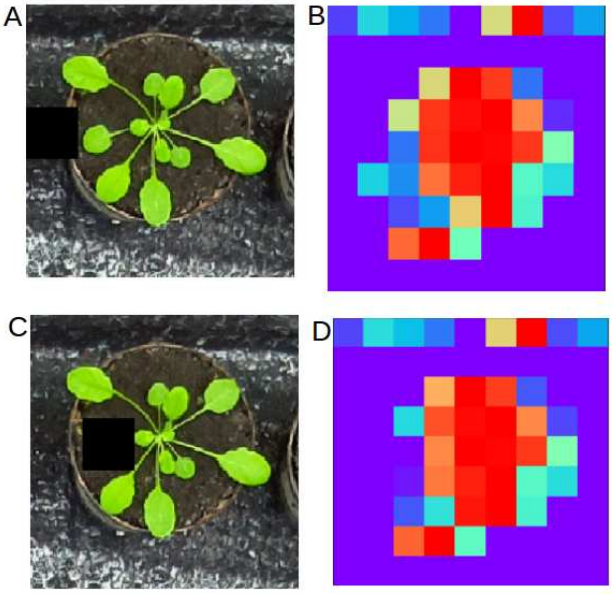


Figure 6. LRP visualization of the last convolutional layer when a leaf is hidden with a black box. The test shows that the network does not 'memorise' a count while ignoring features of the plant: in A, the black box does not hide any leaves and the leaf count prediction is 13, whereas in C the black box hides one of the leaves and the prediction is correctly 12. The heatmaps B and D represent the LRP at the final convolutional layer, showing in D that the black box only has impact in the area where it occludes a leaf.

block and computed the average activation across the feature maps. In the lower part of Figure 5, we show the average output of the last convolutional layer in each block at the beginning and at the end of the training process (only block 2,3, and 5 are displayed for brevity).

The output of the second convolutional block provides low-level features of the input. After the third block, the network starts to focus on edges of the leaves or part of the background. It is the output of the last layer that provides higher activations corresponding to the location of the plant, discarding most of the background. It is worth noting how the convolutional block outputs evolve over time: the second block provides similar features at the beginning and at the end of the training. The output of the third block learns to extract high level features from the blade edges, although edges in the background areas are still very active (e.g., the pots). In the last block the activations start off diffused but after training the network learns how to exclude the background and only keeps meaningful activations in areas that correspond to the plants.

From Figure 5, we can see that the network learns how to mask the background after 5 convolutional blocks (and downsampling operations). In Figure 6, we show an example of the output of the last convolutional block. We noticed that the majority of the highest activations are located in ar-

Feature map	DiC	|DiC|	MSE	%
Unaltered	0.11(1.10)	0.74(0.80)	1.20	41
No top row	0.11(1.08)	0.74(0.80)	1.19	41
Only top row	-9.71(5.73)	9.71(5.73)	126.16	0

Table 2. Test to determine effect on count prediction of the top row of the feature map seen in Figure 6(B,D). "No top row" means that the top row of the feature map is forced to be 0. "Only top row" means we mask every other row except the top row in the feature map to be 0. Results shown are based on the A1-A4 datasets.

eas within the plant. However, we also noted that the top part of the feature maps contains information not related to the plant. This top row is present in all images from the A1-A4 datasets which contain examples of various backgrounds and light intensities. This suggests that the network stores additional information that might not be directly related to the plant features, but it is not evident what this information is related to. Thus, we next investigated whether this information located at the top of the feature map is related to the regression task. We approached this problem in two ways by altering (i) the input; and (ii) the feature map.

Input alteration test: In Figure 6(C), we masked one leaf from the original image (A) and provided the new input to the network. Using LRP, Figures 6(B) and (D) differ only in areas corresponding to the leaf, and the activation information stored on top of the feature map changes by only 2.27% of the activation values, indicating that these activations may not be related to the regression task.

Feature map alteration test: To further demonstrate that the information on top of the feature map in Figures 6(B) and (D) are not related to leaf count, we altered the feature map during testing. Specifically, we added a masking layer after the last convolutional layer in the block 5, where the top row of Figures 6(B) is replaced with 0. Experimental results of all the images in the A1-A4 training sets are shown in Table 2. Even after we replaced the top row with 0's after the last convolutional layer, results show that the leaf count is unchanged and that the DNN can still correctly count leaf numbers. To further confirm it, we performed another experiment, where we masked the central part of the feature map, maintaining the top-row information. As it can be observed in Table 2, in this case the network is unable to make predictions even if the top layer is preserved.

From these experiments we can conclude that:

- VGG-16 focuses on the leaf blades to make leaf counting predictions.
- VGG-16 disregards the surface of the leaves when counting leaves.
- After 5 convolutional blocks, the network focuses on the plant, discarding most of the background.

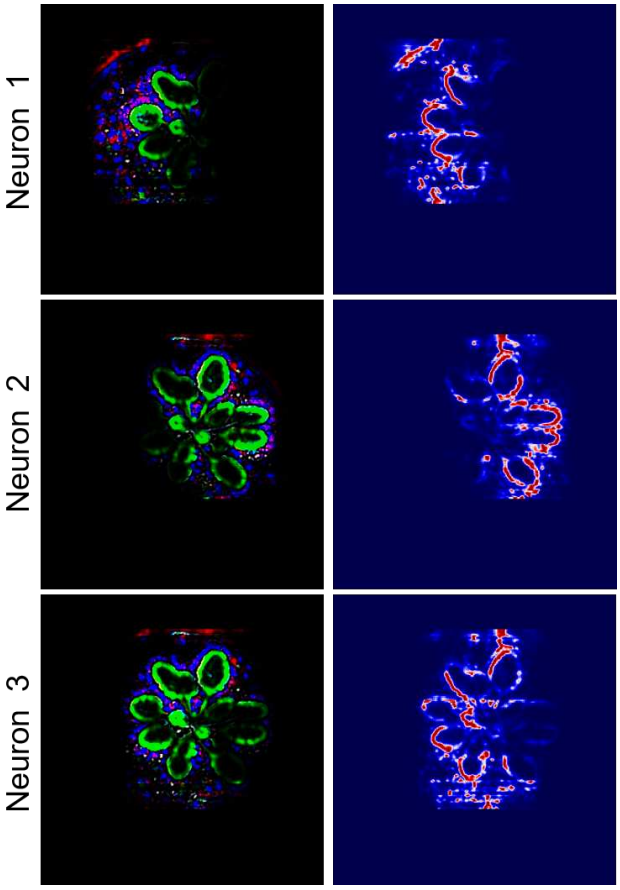


Figure 7. Visualization of the three most influential neurons in the layer following the last convolutional layer (three are shown for brevity). The neurons express an attention, only receiving a signal from part of the input.

- The network stores information *unrelated* to the counting task on top of the last convolutional block.
- The information stored on top of the last convolutional block is unnecessary to predict leaf count.

4.3. What is the regressor looking at?

The last three layers of our VGG-16 network are responsible for learning the regression task. We analyzed the first fully-connected layer, as it is responsible to learn a mapping between the visual features and the predictions. Using LRP, we identified the most active nodes in this 1024-dimensional vector. Focusing on each node in order of relevance we performed guided backpropagation and LRP all the way to the input.

The qualitative results of the three most active nodes are displayed in Figure 7 as an example. Each of these nodes acts as an 'attention map': they get excited by different parts of the plant, ignoring the rest of the image. We hypothe-

Parameters	DiC	|DiC|	MSE	%
57M	0.11 (1.10)	0.74 (0.80)	1.20	41
36M	-0.21 (1.13)	0.79 (0.83)	1.33	41

Table 3. Network compression experiment. We compared the performance of two versions of the VGG-16, which have a different number of parameters.

size that thanks to this 'self-learned attention mechanism' the network is able to distinguish between foreground (the plant) and the background (the soil).

4.4. Compressing the network

When computing the relevance of each neuron in the fully connected layers we found that $\sim 40\%$ of the nodes are mostly inactive irrespective of the input image. This suggests that we could further reduce the size of the fully connected layers. This would reduce computational strain during training and help reduce over-fitting. Fully connected layers are characterized by a weight matrix W with dimensionality depending on their input and output layer sizes. This means that these layers make up a considerable part of parameters.

For this test we halved the number of nodes in the fully connected layers, from 1024 and 512 to 512 and 256, respectively. By making this change the total parameter count decreased from 57 million to 37 million. We trained the reduced parameter network using the same training procedure as the full parameter network. The quantitative results on the full training set are reported in Table 3. We obtained a reduction of the total network parameters by 37%, however the impact in the prediction accuracy is not significant. A paired t-test gives a p-value of 0.17. The MSE prediction error increases by $\sim 10\%$ but the overall agreement remains the same for both networks.

5. Conclusions

Numerous studies have been done to understand the decision making process of DNNs for classification applications in computer vision. However, there has been little analysis into the interpretation of regression based architectures and what the network focuses on to reach a decision. In this paper we addressed the gap by employing deep learning visualization techniques to gain a better understanding of the decision contributing factors in the regression based plant phenotyping task of leaf counting.

We used LRP and guided backpropagation to inspect what areas of the input image are important for the output and we also investigated what information is captured in intermediate layers. We experimentally determined that the blade edge is the most important part of the plant that contributes to the final leaf count, regardless of the image background or plant species and scale. We determined that the network focuses on the plant by observing the activations of intermediate layers during the training process. We show that the network reacts to occlusions of the input and does not store count information in areas not corresponding to the plant. Finally, through improved understanding, we show that we can compress the network, while not significantly impacting the performance.

Acknowledgements

This work was supported by the EPSRC DPT PhD fellowship EP/N509644/1. SAT is supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1. We acknowledge and thank NVIDIA for providing hardware essential for this work.

References

[1] S. Aich, A. Josuttes, I. Ovsyannikov, K. Strueby, I. Ahmed, H. S. Duddu, C. Pozniak, S. Shirtliffe, and I. Stavness. Deepwheat: Estimating phenotypic traits from crop images with deep learning. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 323–332. IEEE, 2018.

[2] S. Aich and I. Stavness. Leaf counting with deep convolutional and deconvolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2080–2089, 2017.

[3] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017.

[4] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[5] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6541–6549, 2017.

[6] J. Bell and H. M. Dee. Aberystwyth Leaf Evaluation Dataset. nov 2016.

[7] D. C. Boyes, A. M. Zayed, R. Ascenzi, A. J. McCaskill, N. E. Hoffman, K. R. Davis, and J. Görlach. Growth stage–based phenotypic analysis of arabidopsis: a model for high throughput functional genomics in plants. *The Plant Cell*, 13(7):1499–1510, 2001.

[8] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[9] J. Choo and S. Liu. Visual analytics for explainable deep learning. *IEEE computer graphics and applications*, 38(4):84–92, 2018.

[10] A. Dobrescu, M. Valerio Giuffrida, and S. A. Tsaftaris. Leveraging multiple datasets for deep leaf counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2072–2079, 2017.

[11] V. Escorcia, J. Carlos Niebles, and B. Ghanem. On the relationship between visual attributes and convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1256–1264, 2015.

[12] M. V. Giuffrida, P. Doerner, and S. A. Tsaftaris. Pheno-deep counter: a unified and versatile deep learning architecture for leaf counting. *The Plant Journal*, 96(4):880–890, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] F. M. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics*, 2018.

[15] Y. Itzhaky, G. Farjon, F. Khoroshevsky, A. Shpigler, and A. B. Hillel. Leaf counting: Multiple scale regression and detection using deep cnns. 2018.

[16] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne. Learning how to explain neural networks: Patternnet and patternattribution. *arXiv preprint arXiv:1705.05598*, 2017.

[17] W. Landecker, M. D. Thomure, L. M. Bettencourt, M. Mitchell, G. T. Kenyon, and S. P. Brumby. Interpreting individual classifications of hierarchical networks. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 32–38. IEEE, 2013.

[18] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition*, 71:1–13, 2017.

[19] Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[20] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[21] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern recognition letters*, 81:80–89, 2016.

[22] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[23] J. Oramas, K. Wang, and T. Tuytelaars. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. *arXiv preprint arXiv:1712.06302*, 2018.

[24] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2017.

[25] H. Scharr, M. Minervini, A. Fischbach, and S. A. Tsaftaris. Annotated image datasets of rosette plants. In *European Conference on Computer Vision. Zürich, Suisse*, pages 6–12, 2014.

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[27] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[31] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[32] Q. Zhang, Y. Nian Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.

[33] Q.-s. Zhang and S.-C. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.